

Table based KNN and AHC Algorithm for combining Text Segmentation with Text Clustering

Taeho Jo
President
Alpha AI Research
Cheongju, South Korea
tjo018@naver.com

Abstract—In this research, we propose the table based KNN algorithm and the table based AHC algorithm for automating the text segmentation by introducing the text clustering. In the previous work, although the KNN version was proposed as an approach to the text segmentation, a domain should be presented as an input text tag in advance, manually. In this research, text clusters are constructed based on their contents through the text clustering, and a text is segmented into subtexts based on their contents, by selecting a classifier which corresponds to its most similar cluster. The AHC algorithm and the KNN algorithm which processes tables are adopted respectively for the text clustering and the text segmentation. The goals of this research are to automate the text segmentation and to improve the performance of both tasks, using the two table-based algorithms.

I. INTRODUCTION

The text segmentation is defined as the process of segmenting logically a text into subtexts, based on their contents. It is necessary to judge whether the topic changes or not between two adjacent paragraphs for segmenting a text. The text segmentation is mapped into a classification task where each adjacent paragraph pair is classified into boundary and continuance. A boundary is established between paragraphs in each pair which is classified into boundary, and a text is segmented by the established boundaries. Subtexts which are outputs of the text segmentation are treated as independent texts.

This research is motivated by the requirement of defining domains manually in designing the text segmentation system. The text segmentation is mapped into a binary classification within each domain; the text segmentation system is composed with the binary classification tasks as many as domains. The process of designing the text segmentation system is to predefine domains by prior knowledge or subjective views, collect sample paragraph pairs domain by domain, and allocate a classifier for each domain. One more motivation of this research is the discovery of the excellent performances in using the table based KNN algorithm and the table based AHC algorithm respectively for the text segmentation and the text clustering. In this research, we connect the text segmentation with the text clustering for automating the domain predefinition and adopt the table

based KNN algorithm and the table based AHC algorithm as the approaches to both tasks.

The idea of this research is to predefine the domains by the text clustering for the text segmentation automatically and apply the table based KNN algorithm and the table based AHC algorithm to both tasks. In this research, we prepare the text collection where each text is segmented into subtexts by the topic transition. We define the similarity metric between two tables, and the KNN algorithm and the AHC algorithm are modified based on the similarity metric into the table-based versions. The collection is segmented into clusters by the text clustering, and the text segmentation system is implemented cluster by cluster. A domain is automatically decided to an input text by its similarities with the cluster prototypes for selecting a corresponding text segmentation system.

Let us mention some benefits from this research. The main problems in encoding texts into numerical vectors are solved by encoding them into other structured forms called tables. The improved performance in the text segmentation which relates to the text clustering is expected by adopting the table-based versions of the KNN algorithm and the AHC algorithm. Because the domain predefinition is automated by the text clustering, it does not require to define domains depending on prior knowledge and subjective views. Because a domain is automatically assigned to an input text based on its similarities with the cluster prototypes, it does not require to tag an input text with its domain manually.

Let us mention the organization of this paper. In Section II, we explore the previous works which are relevant to this study. In Section III, we describe in detail what is proposed in this research. In Section IV, we mention the significances of this research and the remaining tasks. This paper is composed of the four sections.

II. PREVIOUS WORKS

This section is concerned with previous works which are relevant to this research. It is intended to connect the text segmentation with the text clustering for defining domains in a text collection. The direction is to review cases of applying the proposed version of KNN algorithm to the text

clustering and the text segmentation and modifying the KNN algorithm and the AHC algorithm into table-based versions for improving their performances. The difference of this research is that the text segmentation is connected to the text clustering for implementing a complete text segmentation system. This section is intended to explore previous works for providing background for this research.

Let us explore the previous works which are concerned with the application of the modified version of the standard AHC algorithm to text clustering. In 2017, Jo initially proposed that the modified version should be applied to the text clustering [4]. In 2019, Jo validated empirically the table-based AHC algorithm in the text clustering [7]. In 2023, he prepares the journal article which describes the table-based AHC algorithm and its application to the text clustering for its publication [8]. The text clustering system which was implemented in the above works will be connected to the proposed system which is implemented in this research.

Let us explore the previous works on applying the modified version of KNN algorithm to the text segmentation. In 2017, it was initially proposed that the table-based version of KNN algorithm should be applied to the text segmentation by Jo [5]. In 2018, the modified KNN algorithm was validated in the task by Jo [6]. The journal article which describes in detail the modified KNN algorithm and its application to the text segmentation is finally prepared for its publication by Jo [9]. It was pointed out that the text segmentation is a domain dependent classification, and it is necessary to define domains automatically from a particular text collection.

Let us explore the previous works on the table matching algorithm as the early case of encoding a text into a table. In 2008, the table based matching was proposed as an approach to the text classification as the initial case of encoding a text into a table, instead of a numerical vector [1]. In 2011, the table based matching algorithm was registered as a patent by Jo [2]. In 2015, it was upgraded into its more reliable and stable version by Jo [3]. The similarity metric between two tables is provided by the above previous works for modifying the standard KNN algorithm and its variants.

Let us mention the differences of this research from the previous works which are explored above. The text clustering which was implemented in [8] is applied to domain definition for text segmentation. In this research, the domains of text collection are defined by introducing text clustering for upgrading the text segmentation system. The modified version of KNN algorithm becomes more advanced ones, compared with the table based matching algorithm. We will consider introducing text segmentation to text clustering as the reverse case of this research in the next one.

III. PROPOSED WORKS

This section is concerned with what is proposed in this research. In Section III-A, we describe the process of encoding a text into a table and the proposed similarity metric. In Section III-B, we describe the table based version of KNN algorithm. In Section III-C, we present the system architecture of the text segmentation system. In Section III-D, we present the system architecture of its extended version by connecting it with the text clustering system.

A. Encoding Process

This section is concerned with the table as the text representation. It is defined as a set of entries, each of which consists of an element and its weight. The table which represents a text consists of entries, each of which is a pair of a word and its weight in the text. The similarity between tables is computed based on shared words as the semantic similarity between texts. This section is intended to describe the process of encoding a text into a table and the process of computing the similarity between tables.

The process of encoding a text into a table is illustrated in Figure 1. It is indexed into a list of words. The weights of words in the text are computed in the text-word weighting; the word weight in the text is a relationship between a word and a text. The entries with their lower weights are removed for downsizing the table. Refer to [1] for studying the entire process in more detail.

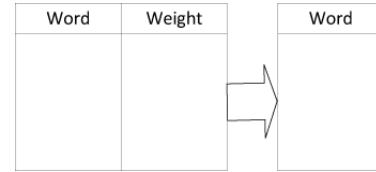


Figure 1. Process of Encoding Texts into Tables

Let us mention the function of a table which maps it into a set of words as shown in Figure 2. The table which represents a text is expressed as entry set as shown in equation (1),

$$T = \{(t_1, w_1), (t_2, w_2), \dots, (t_{|T|}, w_{|T|})\} \quad (1)$$

where t_i is a word, and w_i is the weight of the word, t_i , in the text. The function for generating a list of words is expressed as equation (2),

$$F(T) = \{t_1, t_2, \dots, t_{|T|}\} \quad (2)$$

The function is the operation which takes only words without their weights as a set. The operation is applied to computing the similarity between tables which represent texts.

Let us mention the process of computing the similarity between two tables as the similarity between two texts. The tables which represent texts are notated by the two sets: $D_1 = \{(t_{11}, w_{11}), (t_{12}, w_{12}), \dots, (t_{|D_1|}, w_{|D_1|})\}$ and

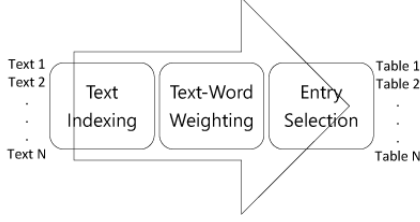


Figure 2. Conversion of Table into Word Set

$D_2 = \{(t_{21}, w_{21}), (t_{22}, w_{22}), \dots, (t_{2|D_2|}, w_{2|D_2|})\}$. The intersection between the two sets which are generated by applying the function, $F(\cdot)$, by equation (3),

$$F(D_1) \cap F(D_2) = \{t_{s1}, t_{s2}, \dots, t_{s|F(D_1) \cap F(D_2)|}\} \quad (3)$$

and the table with entries each of which consists of a shared word, its weight from the table, D_1 , its weight from the table, D_2 , is constructed based on equation (3), as expressed in equation (4),

$$SD_{12} = \{(t_{s1}, w_{s1}^1, w_{s1}^2), (t_{s2}, w_{s2}^1, w_{s2}^2), \dots, (t_{s|F(D_1) \cap F(D_2)|}, w_{s|F(D_1) \cap F(D_2)|}^1, w_{s|F(D_1) \cap F(D_2)|}^2)\} \quad (4)$$

The similarity between the two tables, D_1 and D_2 , is computed by equation (5),

$$sim(D_1, D_2) = \frac{2(\sum_{i=1}^{|F(D_1) \cap F(D_2)|} w_{si}^1 + \sum_{i=1}^{|F(D_1) \cap F(D_2)|} w_{si}^2)}{\sum_{i=1}^{|D_1|} w_{1i} + \sum_{i=1}^{|D_2|} w_{2i}} \quad (5)$$

The value which is computed by equation (5), is always given as a normalized value between zero and one, as shown in equation (6),

$$0 \leq sim(D_1, D_2) \leq 1. \quad (6)$$

Let us make some remarks on the encoding process and the computation of the similarity between texts. A text is encoded into a table by the text indexing, the word weighting, and the entry selection. A table is expressed as a set of entries, and a table is filtered into a set of words by the defined function. The similarity between tables is computed based on shared words by equation (5). In the future research, we consider representing a text into multiple tables.

B. Table based KNN Algorithm

This section is concerned with the table based KNN algorithm. It was initially proposed as the approach to the text classification by Jo in 2015 [3]. The similarity metric between tables which was described in the previous section is used for computing the similarity between a novice table and a training example. The labels of its nearest neighbors are voted with their identical weights for deciding the label

of a novice input. This section is intended to describe the initial version of KNN algorithm from which its variants are derived.

The process of computing the similarity between a novice item and a training example is illustrated in Figure 3. The labeled words which are gathered as samples are encoded into tables, and they are notated by a set $Tr = \{T_1, T_2, \dots, T_N\}$. A novice word is encoded into a table notated by T . Its similarities with the tables which represent the sample words are computed by equation (5), as $sim(T, T_1), sim(T, T_2), \dots, sim(T, T_N)$. Some training examples which are most similar as the novice input are selected as its nearest neighbors.

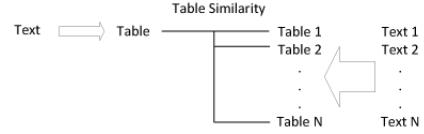


Figure 3. Similarities of Novice Input with Training Examples

In Figure 4, the process of selecting nearest neighbors by the ranking selection is illustrated. The training examples are sorted by the descending order of their similarities, after computing their similarities with a novice example. The training examples with their higher similarities with a novice example are selected as its nearest neighbors, as expressed in equation (7),

$$Nr = Select_k(Tr, T), Nr \subseteq Tr \quad (7)$$

The set of nearest neighbors, Nr , is composed with elements as expressed in equation (8),

$$Nr = \{T_1^{near}, T_2^{near}, \dots, T_k^{near}\} \quad (8)$$

The elements in the set, Nr , are used for voting their labels in the next step.

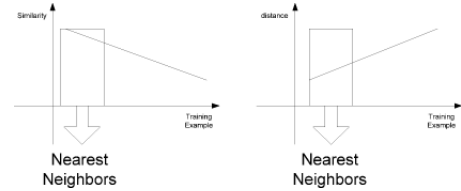


Figure 4. Selection of Most Similar or Least Distant Training Examples as Nearest Neighbors

The process of voting the labels of nearest neighbors for decoding the label of a novice word is illustrated in Figure 5. The predefined categories are notated by $c_1, c_2, \dots, c_m$, and the label of a nearest neighbor is notated by $c_j = label(T_i^{near})$. The number of nearest neighbors which belong to the category, c_j , is notated by $Count(Nr, c_j)$. The label of the novice item, T , is decided by equation (9),

$$label(T) = \arg\max_{j=1}^m Count(Nr, c_j) \quad (9)$$

The process of deciding the label of a novice item by equation (9), is called voting.

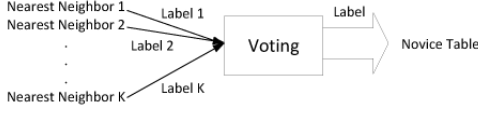


Figure 5. Voting of Labels of Nearest Neighbors for Deciding Label of Novice Input

Let us make some remarks on the table based KNN algorithm. It computes the similarities of a novice item with the training examples. The training examples with their highest similarities with the novice item are selected as its nearest neighbors. The label of the novice item is decided by voting the labels of its nearest neighbors. The ranking selection is adopted for selecting the nearest neighbors from training examples, in this version.

C. System Architecture

This section is concerned with the text segmentation system which adopts the table based KNN algorithm. In Section III-B, we described the proposed version of KNN algorithm as the approach to the text segmentation. It is viewed as a binary classification which classifies a paragraph pair into boundary or continuance. This section is intended to describe the text segmentation system with respect to its function and its architecture.

The process of gathering sample paragraph pairs and classifying a novice one, is illustrated in Figure 6. Because even a same paragraph pair may be classified differently depending on the domain, the task is called domain dependent classification. Sample paragraph pairs are gathered domain by domain, and each pair is labeled with boundary or continuance. The paragraph pairs are taken from texts which are tagged with their same domain, each paragraph pair is classified into boundary or continuance. Sample paragraph pairs which are labeled with one of the two categories are encoded into tables in each domain.

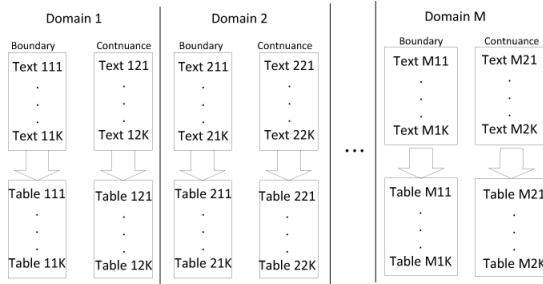


Figure 6. Preparation of Training Examples

The entire architecture of the proposed text segmentation system is illustrated in Figure 7. A text is given as the input, and adjacent paragraph pairs are extracted by partitioning

the text into paragraphs and slide the two sized window on them. The sample paragraph pairs in the boundary group and the continuance group and ones which extracted from the text are mapped into tables in the encoding module. The paragraph pairs from the text are classified into one of the two categories in the similarity computation module and the voting module. A boundary is put between paragraphs in each pair which is classified into boundary in the text, and it is partitioned into subtexts by the boundary.

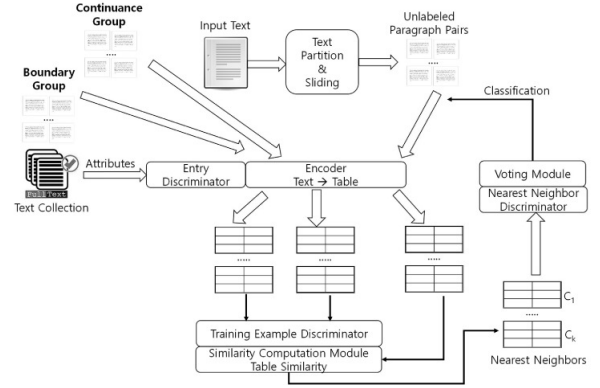


Figure 7. System Architecture

The execution flow of the text segmentation system is illustrated in Figure 8. Texts with same domain are initially collected, adjacent paragraph pairs are extracted from them, and the paragraph pairs are labeled with boundary or continuance. From an input text, adjacent paragraph pairs are extracted, and are encoded into tables, together with the sample paragraph pairs. The adjacent paragraph pairs are classified into boundary or continuance, and there are two groups of paragraph pairs: boundary group and continuance group. The boundary is marked between paragraphs in each pair in the boundary group, and text is partitioned by the marked boundaries into subtexts as the final output of this system.

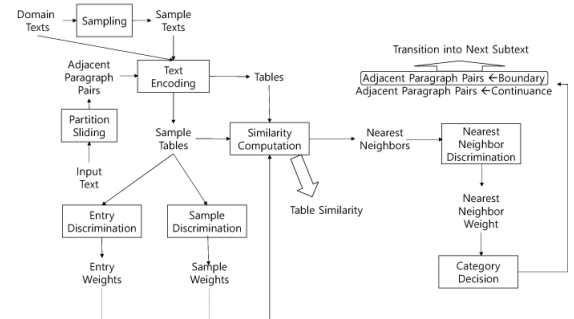


Figure 8. System Flow

Let us make some remarks on the proposed system which is illustrated in Figure 7 as its architecture. The text segmentation is defined as the binary classification where

each adjacent paragraph pair is classified into boundary or continuance. The paragraph pairs are encoded into tables instead of numerical vectors, and they are classified into directly. The input text is partitioned into subtexts by the boundary which is put between paragraphs which are classified into boundary. The subtexts as the semantic division of the input text are treated as independent texts.

D. Connection with Text Clustering

This section is concerned with the connection of the text segmentation with the text clustering. In the previous section, we mentioned the text segmentation as an instance of domain dependent classification. The domains are automatically constructed from the text collection by clustering texts. The AHC algorithm which is proposed in [7] is used for implementing the text clustering. This section is intended to describe the text segmentation system which is connected with the text clustering for automating the construction of domains.

The architecture of the text clustering system which was proposed in [7] is illustrated in Figure 9. The texts in the group are encoded into tables, and the AHC algorithm which is proposed in [7] is adopted as the approach. It starts with singletons as many as items, computes the similarities of all possible cluster pairs, and merge the cluster pair with its highest similarity into one cluster. The two steps are iterated until the number of clusters reach the desired number. We may consider replacing the AHC algorithm by its variants for improving the speed and the performance.

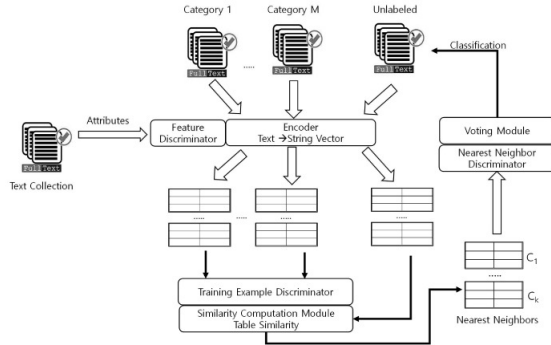


Figure 9. Text Clustering System

The connection of the text segmentation with the text clustering is illustrated in Figure 10. The M domains are defined by clustering texts in a group, initially and automatically. A novice text is given as the input, and its domain, domain D , is decided by its similarities with domains. A classifier which corresponds to domain D is nominated and classify each adjacent paragraph pair in a novice text into boundary or continuance. The M index optimization systems are viewed as independent ones, each of which classifies each paragraph pair into one of the two categories.

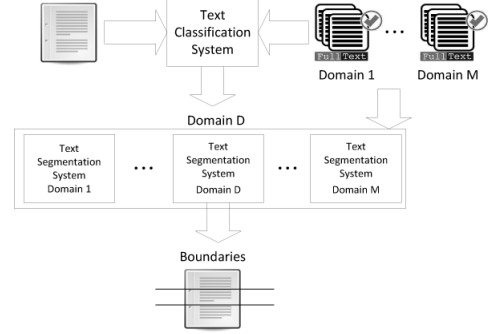


Figure 10. Text Segmentation System connected with Text Clustering

The execution flow of text segmentation which is connected with the text clustering is illustrated in Figure 11. Unlabeled texts are clustered into subgroups, each of which is a domain. Sample paragraph pairs which are labeled with boundary or continuance are generated from each domain. These sample paragraph pairs are provided for training the classifier in each text segmentation system. Each classifier is used for judging whether the topic change happens between paragraphs in each pair, or not.

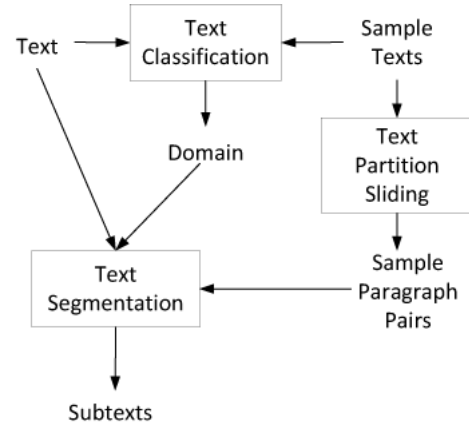


Figure 11. System Flow of Text Segmentation connected with Text Clustering

Let us make some remarks on the connection of the text segmentation with the text clustering. It is the process of segmenting a text group into subgroups based on their content-based similarities. The role of text clustering is to define the domains initially in the architecture of connecting the text segmentation with the text clustering. In each domain, sample paragraph pairs are generated, and its corresponding classifier is trained with them. In the future research, we consider using the text segmentation for clustering texts in the opposite direction.

IV. CONCLUSION

Let us mention the significances of this research as the conclusion. We modify the KNN algorithm into the table

based version. We design the text segmentation system by adopting the proposed KNN algorithm. We connect the system with the text clustering system for automating the domain decision. In the next research, we will implement the text segmentation system as a real system.

Let us mention some remaining tasks as the further discussions on this research. More advanced machine learning algorithms and deep learning algorithms are modified into table-based versions. We need to define and characterize mathematically additional operations on tables. We implement the text segmentation system which relates to the text clustering as a real program or a library. The text classification and the text clustering relate to the text segmentation for improving their performances.

REFERENCES

- [1] T. Jo, "Table based Matching Algorithm for Soft Categorization of News Articles in Reuter 21578", 875-882, Journal of Korea Multimedia Society, 11, 2008.
- [2] T. Jo, "Device and Method for Categorizing Electronic Document Automatically", 10-2009-0041272, 10-1071495, 2011.
- [3] T. Jo, "Normalized Table Matching Algorithm as Approach to Text Categorization", 839-849, Soft Computing, 19, 2015.
- [4] T. Jo, "Table based AHC for Text Clustering", 133-138, The Proceedings of 13th International Conference on Data Mining, 2017.
- [5] T. Jo, "Content based Segmentation of Texts using Table based KNN", 27-32, The Proceedings of 16th International Conference on Advances in Information and Knowledge Engineering, 2017.
- [6] T. Jo, "Using Table based Version of K Nearest Neighbor for Segmenting News Articles by their Contents", 62-64, The Proceedings of 1st International Conference on Advanced Engineering and ICT-Convergence, 2018.
- [7] T. Jo, "Applying Table based AHC Algorithm to News Article Clustering", 8-11, The Proceedings of International Conference on Green and Human Information Technology, Part I, 2019.
- [8] T. Jo, "Similarity Metric between Tables for Modifying AHC Algorithm in Text Clustering", in Preparation, 2023.
- [9] T. Jo, "Segmenting Long Texts Automatically by Table based K Nearest Neighbor", in Preparation, 2023.